Lab 1 – Introduction to Ansible

I- Server configuration and remote machines:

1- Create an inventory file “hosts” in your working directory.
   a. Add localhost local root user.
   b. Choose two hosts of your colleagues' VPS with the root user:
      The one who precedes you and the one who follows you having respectively the aliases **avant** and **apres**.

```
root@vps290950:~# cd /root

root@vps290950:~# touch inventory

root@vps290950:~# vi inventory

root@vps290950:~# cat /root/inventory

root@vps290950:~# cat inventory
[local]
127.0.0.1
[avant]
5.196.15.248
[apres]
5.196.15.248

[local:vars]
ansible_ssh_user=root
ansible_ssh_pass=****

[all:vars]
ansible_connection=ssh
```

2- Do the necessary to copy the SSH keys from each host.
   Add public key for your host in authorized keys in “avant” et “après”:

   > ~/.ssh/authorized_keys

II- Ad-hoc Commands:

1- Test the connection to all remote clients.

   >ansible all -i inventory -m ping

```
root@vps290950:~# ansible all -i inventory -m ping
The authenticity of host '5.196.15.248 (5.196.15.248)' can't be established.
RSA key fingerprint is SHA256:0RazCX19fnAiBSDRGNehJfGIcy9nHjU0o6W3wW/swJY.
Are you sure you want to continue connecting (yes/no)? yes
5.196.15.248 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Warning: Permanently added '5.196.15.248' (RSA) to the list of known hosts.\r\
nPermission denied (publickey,password).",
    "unreachable": true
}
```

```
root@vps290950:~# ansible all -i inventory -m ping
[WARNING]: Unable to parse /root/inventory as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
```

```
root@vps290950:~# ansible all -i inventory -m ping
[WARNING]: Platform linux on host 5.196.15.248 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
5.196.15.248 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

2- Retrieve all information from the remote machine **avant**.
> ansible <hostname> -m setup

```
root@vps290950:~# ansible avant -i inventory -m setup
[WARNING]: Platform linux on host 5.196.15.248 is using the discovered Python interpreter at /usr/bin/python, but future
installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
5.196.15.248 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "5.196.15.248"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::f816:3eff:fe5f:ec7e"
        ],
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "04/01/2014",
        "ansible_bios_version": "2:1.10.2-58953eb7",
        "ansible_bond0": {
            "active": false,
            "device": "bond0",
            "lacp_rate": "slow",
            "macaddress": "c6:e0:83:56:b2:bd",
            "miimon": "0",
            "mode": "balance-rr",
            "mtu": 1500,
            "promisc": false,
            "slaves": [],
            "type": "bonding"
        },
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/bzImage-3.14.32-xxxx-grs-ipv6-64-vps",
```

3- It is not safe to keep an ssh connection to the root user ; use the module **user** to create an **«ansible»** user in the machines **avant** and **apres** with an ansible password.

> mkpasswd --method=sha-512
password=welcome0

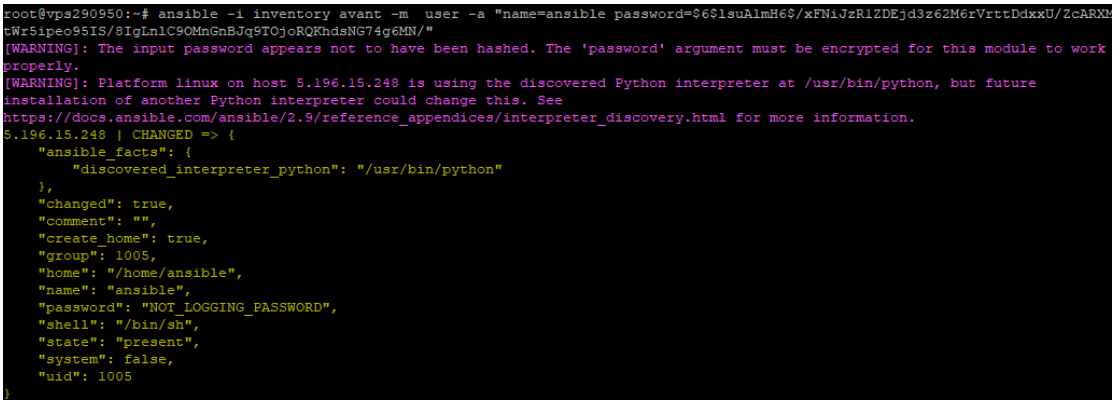
```
root@vps189595:~# mkpasswd --method=sha-512
Password:
$6$lsuAlmH6$/xFNiJzR1ZDEjd3z62M6rVrttDdxxU/ZcARXMtWr5ipeo95IS/8IgLnlC9OMnGnBJq9TOjoRQKhdsNG74g6MN/
```

Or

```
> ansible all -i localhost, -m debug -a "msg={{ 'mypassword' | password_hash('sha512', 'mysecretsalt') }}"
```

NB : the user module password argument you find in the documentation must contain the value of the hashed password that will be directly inserted into /etc/shadow, check the documentation of the user module to know how to encrypt the password.

>ansible -i inventory avant -m user -a "name=ansible password=6lsuAlmH6$/xFNiJzR1ZDEjd3z62M6rVrttDdxxU/ZcARXMtWr5ipeo95IS/8IgLnlC9OMnGnBJq9TOjoRQKhdsNG74g6MN/"



Final solution :

*** Create user-ansible on nodes**

> ansible localhost -i inventory -m debug -a "msg={{ 'passforce' | password_hash('sha512', 'sceretsalt') }}"

> ansible -i inventory -m user -a 'name=user-ansible password=6sceretsalt$tBcfGEgifQpQZsg5CIGZ79XC55h5vHy7UWrys7cAF37KNCQQbm7iCvy58MlLQLaS2fLF6ZjqDVHhVrkMdRi0f0' --user root --ask-pass all

*** Give sudo rights to user-ansible:**

> ansible -i inventory -m user -a 'name=user-ansible groups=wheel append=yes ' --ask-pass --become --ask-become-pass all

4- Remove the user's indication in the inventory and now indicate in the command your target host «ansible».

> ansible -i inventory -m ping localhot --user ansible --ask-pass

5- Copy a file from your local machine to the home directory of Ansible.

>ansible -i inventory avant -m copy -a "src=/root/create_user.yml dest=/home/ansible" --user=ansible

ansible -i inventory avant -m copy -a "src=/root/create_user.yml dest=/home/ansible“

6- Run the **ls** command to check that the file has been copied (always via Ansible of course).

>ansible precedents -i inventory -m file -a "path=/home/ansible/create_user.yml state=file"

7- Now try to copy the file into **/var/log :**

>ansible suivants -i inventory -m copy -a "src=/root/create_user.yml dest=/var/log" --user ansible --ask-pass

a. Is the command successful?
   If not, try to re-execute the command with the ansible user but as root (via sudo), what option is made for this?

>ansible localhost -m copy -a "src=/root/create_user.yml dest=/var/log" --user ansible --become

b. Make the necessary for the command to be successful

8- Install the **httpd** package in the **avant** machine.
> ansible avant -m yum -a "name=httpd state=present"

```
root@vps290950:~# ansible localhost -m apt -a "name=apache2 state=present"
localhost | SUCCESS => {
    "cache_update_time": 1606681005,
    "cache_updated": false,
    "changed": false
}
```

9- It's usually not a good idea to install a package directly.
You must first update the package database (download it from the repository) to request a version of a package that still exists. Install the emacs package on the **avant** machine by updating the database.

>ansible localhost -m yum -a "name=emacs state=latest update_cache=yes"

```
root@vps290950:~# ansible localhost -m apt -a "name=emacs state=latest update_cache=yes"
localhost | SUCCESS => {
    "cache_update_time": 1606681501,
    "cache_updated": true,
    "changed": false
}
```

10- Check that both packages are installed using the module **command**.

>ansible localhost -m command -a "apt -qq list apache2 emacs "

```
root@vps290950:~# ansible localhost -m command -a "apt  -qq list apache2 emacs "
localhost | CHANGED | rc=0 >>
apache2/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.17 amd64 [installed]
emacs/xenial,now 46.1 all [installed]
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

11- Copy a file from your local server to/var/www/html on the machine **avant**.

12- Try to access the server from your laptop's browser.

13- If successful, uninstall the **httpd** package.

> ansible localhost -m command -a "yum remove httpd "

```
> ansible localhost -m yum -a "name=httpd state=absent"
```

14- Create two groups in your inventory: **précédents** and **suivants** containing respectively the two preceding and two following machines.

```
echo "[précédents]" > inventory
echo "host01 ansible_ssh_user=ansible" >> inventory
echo "host02 ansible_ssh_user=ansible" >> inventory
echo "[suivants]" >> inventory
echo "host03 ansible_ssh_user=ansible" >> inventory
echo "host04 ansible_ssh_user=ansible" >> inventory
```

15- Test connection to group **précédents**.

```
>ansible all -i inventory -m ping --limit precedents
>ansible precedents -i inventory -m ping
```

16- Uninstall **emacs** on the machines of the group **suivants.**

```
>ansible suivants -i inventory -m command -a "yum remove httpd "
```

17- Verify that **emacs** is properly uninstalled.

```
>ansible suivants -i inventory  -m yum -a "name=httpd state=absent"
```